CS 301 Languages and Automata Practice Assignment For Final (Fall 2012)

1. Recall that we define a Turing machine M as a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$. What does each element of this tuple refer to?
2. Is every recognizable language decidable? If YES, sketch a proof why. If NO, give an example of a problem which is recognizable, but not decidable.
3. What is the difference between an NP-hard and NP-complete language?
4. We believe the Turing machine model captures everything a physically realizable computer could do. Which famous statement from class embodies this belief?
5. Prove that if a language is described by a regular expression, then the language is regular.
6. Prove that if a language is context-free, then some pushdown automaton recognizes it.
7. One of the most famous open questions in theoretical computer science is whether $\mathrm{P} = \mathrm{NP}$. Interestingly, however, if we don't care about efficiency of simulation, then the set of languages recognizable by a deterministic Turing machine is the same as the set of languages recognizable by a non-deterministic Turing machine. Prove this fact.
8. Prove that the language $E_{TM} = \{\langle M \rangle \mid M \text{is a TM and } L(M) = \emptyset\}$ from Section 5.1 is undecidable.
9. Recall the notion of a 2-tape TM M, which has two infinite-length tapes and one head per tape. Prove that a single-tape Turing machine can simulate any computation on a 2-tape Turing machine.
10. Consider language $A_{CFG} = \{\langle G, w \rangle \mid G \text{ is a CFG that generates } w\}$ from Section 4.1. Show that A_{CFG} is decidable. Hint: Think Chomsky Normal Form!
11. Let B denote the set of all infinite sequences over $\{0, 1\}$. Show that B is uncountable using a proof by diagonalization.
12. [6 marks] Consider an algorithm A which takes as input $\langle G, c \rangle$ for G a graph on n vertices and c an integer. Suppose that on any such input, A halts and outputs accept or reject in at most T steps. Which of the following values for T are considered "polynomial time"? Justify your answers briefly. You may assume the encoding size of the graph G is n^2 bits.
    (a) $T \in O(n^{100})$
    (b) $T \in O(n^{1000} \log^2 c)$

(c) $T \in O(n^2\sqrt{c})$

13. Recall that we defined the class NP in two ways:

    (A) NP is the class of languages decidable by a polynomial-time non-deterministic TM.

    (B) NP is the class of languages L for which whenever input $x \in L$, there exists a polynomial-size proof y convincing an efficient deterministic verifier V of this fact.

    (a) Sketch why if a language L has an efficient verifier V, then L is decidable by a polynomial time non-deterministic Turing machine.

    (b) Prove that the language $CLIQUE = \{\langle G, k\rangle \mid G \text{ has a clique of size at least } k\}$ is NP-hard.